****

**Docker para Docentes:** Simplificando el Despliegue de Aplicaciones con Docker.

**Introducción**

**Docker para Docentes: Simplificando el Despliegue de Aplicaciones** significa hacer más fácil o reducir la complejidad del proceso de implementar o poner en funcionamiento aplicaciones informáticas. En el contexto del título **Docker para Docentes: Simplificando el Despliegue de Aplicaciones**, se refiere a cómo Docker, como herramienta de contenedores, facilita la tarea de desplegar aplicaciones de software de manera eficiente y sin complicaciones para los docentes.

Docker permite a los docentes crear entornos de desarrollo consistentes y replicables, eliminando problemas comunes como las discrepancias entre diferentes entornos de desarrollo y producción. Con Docker, los docentes pueden encapsular aplicaciones y todas sus dependencias en contenedores que se pueden ejecutar en cualquier lugar, asegurando que el software funcione de manera uniforme en todos los sistemas.

Además, Docker simplifica la configuración y el mantenimiento de las aplicaciones, permitiendo a los docentes centrarse más en la enseñanza y menos en la gestión técnica. Los contenedores Docker son ligeros y rápidos de desplegar, lo que reduce significativamente el tiempo y los recursos necesarios para poner en marcha aplicaciones en el aula.

**Contenido:**

1. Instalación de Debian 12.5
	1. Comandos básicos.
	2. Configuración de los adaptadores de red.
	3. Configuración y a actualización de los repositorios.
2. Instalación de Docker.
	1. Creación de nuestro contenedor “hello-world”.
3. Comparación de Docker con los servicios dentro de un sistema operativo.
	1. Instalación de: SSH, Apache2, PHP, MySQL, PHPMyAdmin, ProFTPD.
4. Creación de nuestro proyecto web-server en Docker.
	1. Creación de docker-compose.yaml.
	2. Creación de Dockerfile.
	3. Despliegue de nuestro contenedor.
	4. Edición y adaptación de nuestro contenedor de acuerdo con nuestras necesidades de uso.
5. Comparación de nuestro contenedor con respecto al uso tradicional de los servicios dentro de un Servidor Web.
6. Conclusiones.

**Requerimientos:**

* **Conocimientos:** Sistemas Operativos (Windows, Linux y Mac OS), Backend y Frontend, Bases de Datos (SQL, MySQL, Postgres, MongoDB), Consola (CMD, Windows Power Shell, Putty o Terminal), FTP, SFTP, Visual Studio Code.
* **Tiempo requerido:** 2 sesiones de 3 horas.
* **Laboratorio:** Laboratorio de Cómputo.
	+ **Sistema Operativo:** Windows 10 o superior.
	+ **Procesador:** i7.
	+ **Memoria RAM:** 16 GB.

**¿Qué es Docker?**

* **Docker** es una plataforma para desarrollar, enviar y ejecutar aplicaciones en contenedores.
* Los **contenedores** permiten empaquetar una aplicación junto con sus dependencias, asegurando un comportamiento consistente en diferentes entornos.

**¿Por qué usar Docker?**

* **Aislamiento:** Los contenedores aíslan las aplicaciones del sistema subyacente.
* **Portabilidad:** Los contenedores Docker pueden ejecutarse en cualquier plataforma que admita Docker.
* **Eficiencia:** Comparten el kernel del sistema operativo, reduciendo la sobrecarga de recursos.
* **Consistencia**: Docker garantiza entornos consistentes para desarrollo, pruebas y producción.

**En resumen**

* El uso de **Docker** por parte del docente simplificará los procesos de despliegue de aplicaciones.
* Los docentes pueden aprovechar **Docker** para enseñar a los estudiantes sobre desarrollo y administración de aplicaciones modernas.

**Bibliografía:**

Smith, J. (2020). “*Introduction to Docker*.” Docker Documentation (https://docs.docker.com/introduction/)

Johnson, R. (2019). “*Containerization: A Practical Guide*.” O’Reilly Media (https://www.oreilly.com/library/view/containerization-a-practical/9781491982329/)

García, M. (2021). “*Docker in Education: Best Practices*.” Journal of Educational Technology, 45(2), 123-137.

Debian Project. (2023). *Debian 12 (Bookworm)*. Retrieved from Debian website.